I. J. Computer Network and Information Security, **2017, 1, 54-63**
Published Online January 2017 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2017.01.07

Analysis of User Identity Privacy in LTE and Proposed Solution

Abdulrahman A. Muthana
Thamar University/Faculty of Computer Science and Information Systems, Thamar, 9676, Yemen
E-mail: ab.muthana@smartsecurity-y.com

Mamoon M. Saeed
Yemen Academy for Graduate Studies, Sanaa, 9671, Yemen
E-mail: mamoon530@gmail.com

Abstract—The mechanisms adopted by cellular technologies for user identification allow an adversary to collect information about individuals and track their movements within the network; and thus exposing privacy of the users to unknown risks. Efforts have been made toward enhancing privacy preserving capabilities in cellular technologies, culminating in Long Term Evolution LTE technology. LTE security architecture is substantially enhanced comparing with its predecessors 2G and 3G; however, LTE does not eliminate the possibility of user privacy attacks. LTE is still vulnerable to user identity privacy attacks. This paper includes an evaluation of LTE security architecture and proposes a security solution for the enhancement of user identity privacy in LTE. The solution is based on introducing of pseudonyms that replace the user permanent identifier (IMSI) used for identification. The scheme provides secure and effective identity management in respect to the protection of user privacy in LTE. The scheme is formally verified using proVerif and proved to provide an adequate assurance of user identity privacy protection.

Index Terms—LTE (Long Term Evolution), IMSI (International Mobile Subscriber Identifier), Anonymity, AKA (Authentication and Key Agreement), User Privacy.

I. Introduction

Recently protecting user identity privacy in cellular networks has received an increasing interest more particularly in Long Term Evolution (LTE) cellular technology. LTE cellular technology, which is recently proposed by the Third Generation Partnership Project [1], has security enhancements comparing to its predecessors: Global System Mobile Communication (GSM) and Universal Mobile Telecommunications System (UMTS). LTE security architecture is substantially different from its predecessors in GSM and UMTS and offers a range of security features.

To protect the user identity privacy, the LTE allocates various different temporary identities such as Global User Temporary Identifier (GUTI), temporary mobile subscriber identifier (TMSI), and cell radio network temporary identifier (C-RNTI) to a single user equipment (UE) at different levels of LTE network architecture for different services. The UE can use these identities instead of the International Mobile Subscriber Identifier (IMSI) to identity itself. This strategy aims at eliminating the IMSI exposure problem and mitigating user identity privacy attacks.

Despite this security strategy the LTE still has a number of security flaws [2-6]. The user identity is still vulnerable to privacy attacks. There are some occasions when the UE is requested to identify itself with its permanent identifier IMSI. Such situations occur when the network fails to retrieve the temporary identifiers of the UE and asks the UE to transmit the UE's IMSI, which in turn transmits it (in clear text) [2]. The IMSI can be intercepted by an attacker, who then could track the movements of the user, and thus violating the user's privacy.

In this paper we analyze the user identity privacy issues in LTE with a focus on authentication and key agreement AKA protocol in LTE. We also present a solution for enhancing AKA protocol and protecting user identity privacy. The solution provides a high level of user anonymity within LTE network through introducing of pseudonyms that replaces the user permanent identifier (IMSI). User identity privacy is preserved with minimal modifications at network architecture. The proposed solution design strategy aims at keeping the messaging system away as much as possible from the modifications and changes. We believe that this solution could be fit easily in current cellular network architecture.

Our main contribution is the demonstration how a particular realization of an existing normal protocol employed by LTE such as AKA protocol can be obtained that substantially enhances the user identity privacy preserving capability in LTE. The privacy enhancement is obtained with minimal changes on the network entities (i.e., Home Subscriber Server HSS and UE) and with no changes in the message system. The second contribution is an extensive theoretical study on user identity privacy in LTE.

The remainder of this paper is organized as follows: Section 2 describes AKA authentication protocol and user privacy issues in LTE. A summary of related work is

given in Section 3. Section 4 and 5 present the proposed solution and its security analysis and Section 6 concludes.

II. User Privacy Issues in LTE

LTE falls short to protect user privacy under all circumstances [3]. Malicious third parties pose challenges on the LTE system to protect the privacy of subscribers from risks associated with knowing subscribers identities. The following sections describe AKA authentication protocol and user identity privacy issues within the LTE network.

A. AKA Protocol

The AKA protocol provides mutual authentication and establish a shared cipher key CK and integrity key IK between the UE and the home network (HE) [1]. Each UE and the HSS share a long-term secret key K. Two sequence numbers SQNHE and SQNUE are also maintained by the HSS and the UE respectively to support the network authentication. The sequence number SQNHE is an individual counter for each user, which is used in the HSS for the generation of AVs while the sequence number SQNUE is the highest sequence number the USIM (Universal Subscriber Identity Module) has accepted. AKA makes use of a set of message authentication code functions {f1, f2} and key generation functions {f3, f4, f5}. Results of all functions are encrypted using K.

When the UE wishes to attach to a network, it transmits to the Mobility Management Entity (MME) an attach request that includes the user permanent identity IMSI. Upon the receipt of the UE's attach request, MME invokes AKA procedure by transmitting to the HSS authentication data request that includes the IMSI. Once HSS receives the authentication data request from MME it responds by generating the authentication vector AV. First, it generates unpredictable random challenge RAND (128 bits) and then computes the sequence number SQN. Next, the message authentication code MAC is computed over SQN, authentication management field AMF, and RAND using f1. After that it computes the expected response XRES, the ciphering key CK, the integrity key IK, and the anonymity key AK over RAND using f2, f3, f4, and f5 respectively. The authentication token AUTN that consists of the sequence number SQN XORed with the anonymity key AK, AMF, and MAC is created. Finally, the HSS creates the authentication vector AV that includes AUTN, RAND, XRES, CK, and IK. The HSS sends AV to MME that keeps XRES and forwards RAND and AUTN to the UE within an authentication request.

Upon receipt of RAND and AUTN the UE verifies the AUTN token and computes the authentication response message. The UE first computes the anonymity key AK using f5 and retrievers the sequence number XSQN by XORing the protected SQN included in AUTN with AK. The UE then computes XMAC over XSQN, AMF and RAND and compares XMAC with the MAC included in the AUTN. If MAC verification passes, the UE verifies the received sequence number XSQN. If the XSQN is within the correct range, the UE computes the response RES, the ciphering key CK and the integrity key IK. Finally, it transmits back the response RES to the MME and stores the two keys in the phone. The MME compares the response of the UE (RES) with the expected response XRES that was included in AV received from HSS. If they are equal, the authentication is successful; otherwise the MME sends an authentication reject message to the UE.

B. User Identity Privacy Issues

The main privacy issue within the LTE is the exposure of the IMSI. The IMSI can be intercepted by adversaries. Such attacks are commonly referred to as "IMSI catching" [13]. To protect the user identity privacy, the LTE allocates various different temporary identities such as GUTI, M-TMSI, and C-RNTI to a single UE within LTE network architecture for different networking services. The UE can use these identities instead of the IMSI to identity itself to the network to initiate service requests.

Despite this security management the user identity is still vulnerable to privacy attacks. In some events the UE needs to identify itself with its IMSI (in clear text) [2]. Some Examples include: (1) During the initial attach procedure; (2) Whenever the serving network (MME) cannot retrieve the GUTI of the UE; (3) Whenever the UE attaches to a new MME that fails to acquire the UE's information from the old MME. Furthermore, it is shown that the use of temporary identifiers like TMSIs is not sufficient to prevent IMSI exposure. TMSIs remain valid for too long and re-used over different areas and can be used in passive attacks against IMSI [16].

C. Location Privacy Issues

IMSI Catching is closely related to the issue of location privacy. Knowing IMSI of user allows the attacker to track the user's movements and create profile about the user and thus breaches the user’s location privacy [15]. Similarly, invading location privacy threatens user identity privacy. IMSIs could be caught at specific locations like shop fronts, where the shop keepers collect IMSIs through WiFi signals for the purpose of evaluating the effectiveness of their shop fronts [17]. Also, location based services (LBS), which have the ability to locate a mobile user geographically can threatens the privacy and the security of the users [18]. The caught IMSIs of a particular user at a specific location could be then used to invade users’ privacy and monitor the users at other locations as specified by the attacker to perform dangerous attacks against [19, 20].

III. Related Work

The research community has put efforts to enhance AKA protocol and user identity privacy. Public key cryptography has been used to encrypt the IMSI in [7-9]. The UE uses the public key of the HSS to encrypt its IMSI and no one could be able to decrypt it except the HSS. In this way, the anonymity of the UE would be

preserved during the attach procedure. Although public key cryptography offers anonymity of the user, the high computation complexity and communication overhead as a result of transmitting encrypted IMSIs from one MME to another in handover and authentication procedures are still considerable.

The authors in [7], with focus on providing user confidentiality with minimum cost, propose Ensured Confidentiality AKA protocol (EC-AKA) to add confidentiality and integrity to the authentication mechanism

In [8] the authors focus on improving the security of the AKA protocol. They focus on weaknesses in the sequence number mechanism and the need to occasionally disclose IMSI. They analyze the deficiencies of the AKA protocol, and propose a Security Enhanced Authentication and Key agreement (SE-EPS-AKA) protocol based on Wireless Public Key Infrastructure using the ECC (Ellipse Curve Cipher) encryption. The proposal avoids the sequence number mechanism exist in AKA protocol and adopts independent serial number (SQN) management mechanism to avoid the failure phenomenon of SQN pseudo synchronization [10].

The authors in [9] propose the adoption of a lightweight public key infrastructure (PKI) providing each mobile network operator (MNO) with a private/public key pair. The public key of a network provider can be stored in the USIM. The UE can encrypt the IMSI with the public key and deliver it to the network in a confidential manner. The approach does not require a public/private key pair to be assigned to the UEs.

In [11] the author aims at providing mutual entity authentication in LTE. The author proposes a new 3-party online authentication protocol, with full mutual authentication between the UE and the HSS. Similar to [8], the proposal avoids the use of the sequence number mechanism. The solution modifies the message elements: new message elements are included and the authentication token (*AUTN* field) is redefined. It also requires replacing the USIM with a new subscriber module ESIM. This is a considerable operational downside.

In [2] the authors specifically attempt to improve UIC in LTE. The proposed scheme is based on what is called a Dynamic Mobile Subscriber Identity (DMSI), which is composed of *MCC||MNC||RIC||ERIC*. The *RIC* and *ERIC* are a random identity confidentiality number and the encrypted RIC. The *RIC* and *ERIC* both need to be 128 bit wide. The *RIC* (and *ERIC*) are dynamically assigned and the association with IMSI is unknown to external parties. An advantage of the approach is that the AKA signaling schemes don't need much modifications for the scheme to work, although the identity presentation would have to change considerably. However, the scheme has some shortcomings: On one hand, it is relatively complex with several new cryptographic functions. The management of RICs would need additional processing effort and memory cost. In addition, the amount of computational effort for mobile devices is high since DMSI is changing with every authentication procedure.

The authors in [12] propose the introduction of an anonymous dummy IMSI to be used during the initial presentation that precedes location registration. The MME will recognize that the IMSI is a dummy IMSI, and it will therefore initiate the ID REQ procedure to retrieve the true IMSI. This is a standard procedure, normally used when the network fail to recognize the presented temporary identity (GUTI). The UE will reply with an ID RES in which the IMSI is encrypted by means a public Identity-Based Encryption (IBE) key.

The authors in [3] suggest concealing the real IMSI within a random bit stream of certain size where only the subscriber and HSS could extract the respective IMSI. The IMSI bits are being substituted with randomly selected bits of a hashed value in a way that could be extractable by HSS. The subscriber, upon receiving an identity request from MME, will randomly select an initial pattern and a key from their respective encrypted memory for the current session. First, the chosen key, Ks, and the initial pattern, Ps, will be decrypted via AES decryption algorithm. Then, the HMAC result of the concatenation of Ps, the current time, Tc, as a time stamp and a nonce, n, will be calculated using the key. HKs; Ps = HMAC(Ps||n||Tc) Ks.

Another method by the same authors is also proposed. The proposal requires that HSS and subscribers have some long-term shared keys and symmetric encryption algorithms are used in order to encrypt IMSIs. In this case, the user selects a random key, Ks, and a random pattern, Ps, along with their respective index values. Then, the user pads IMSI with the pattern as follows: IMSI' = IMSI||Ps. The user then encrypts IMSI' with that key using a symmetric encryption algorithm like AES and sends the message as the following: Mir = E(IMSI') ||Tc||Kc||Pc. Where Tc is the current time, Kc is the key index value and Pc is the initial pattern index value. The HSS upon receiving Mir, fetches the same key and pattern based on the indexes, decrypts the IMSI' and recovers the IMSI value. For the proposed protocols to be working properly and securely, the bit length of the initial patterns and keys should be chosen carefully.

The authors in [13] propose a solution where the IMSI is replaced with a changing pseudonym that only the home network can link to the user's identity. This hiding of the IMSI is done without changing any of the system messages. During authentication, the HSS supplies the UE with a random new IMSI, which referred to as Pseudo Mobile Subscriber Identifier (PMSI). The UE uses the new PMSI the next time it is requested to reveal its IMSI. They propose to use the random challenge (RAND) to provide the UE with the PMSI. Their solution requires that HSS must be extended to store three additional values for each UE: the new shared secret key κ and the two PMSI values p and p′. Here p is used to store the PMSI value the UE's is currently using and p′ stores the new PMSI value that the HSS server designates as the successor PMSI for that UE. A provider implementing this solution would change the normal routine of its HSS server, when composing an authentication request for PMSI.

IV. The Solution

The idea is to replace the permanent identity IMSI of the UE with a temporary identity (Changing Mobile Subscriber Identity (CMSI)) that only the HSS server can map it the UE's IMSI. The UE shall transmit the CMSI when it is requested to present its IMSI. Since only the UE and the HSS know about the UE's IMSI, the privacy of the user identity is preserved.

During authentication, a fresh unpredictable CMSI called CFRESH is generated by the HSS and confidentiality transmitted to the UE. In this regard, we propose changes in the characteristics and uses of some original authentication parameters mainly RAND and SQN. We propose to use the challenge RAND to supply the UE with the new CMSI and the sequence number SQNHE and to use SQN token as a key to encrypt the RAND challenge. SQN is now a key that is randomly generated at each run of the enhanced AKA protocol. The sequence number is delivered to the UE using the token SQNHE, which is securely embedded within the RAND challenge. The UE extracts the new CMSI (CNEW) and the sequence number SQNHE from RAND during a normal verification process. If the verification process passes, the UE updates its CMSI to the new CMSI (CNEW) and uses it in the next time when it needs to identify itself with its IMSI.

The enhancement of AKA protocol requires changes in the HSS and the UE as well as in the procedures of handling authentication requests at the HSS and the UE. No changes are made in the exchanged messages. The changes are illustrated in Fig.1 and Fig.2 and discussed in details below.

A. The HSS

The HSS is extended to store two CMSI values C and CNEW (34 bits each) for each UE. C is used to store the active CMSI currently in use by a UE while CNEW stores the newly generated CMSI, which is allocated to the UE to use the next time it needs to present its IMSI. The HSS stores the additional values C and CNEW in its database against the UE's IMSI and the secret key K (Fig.1). This arrangement ensures that the HSS can always link the currently active CMSI stored at the UE with the corresponding IMSI at the HSS and uniquely identify the UE. Similarly, MME also maintains C and CNEW in its database for each UE within its service area in order to be able to uniquely identify the UE.

CMSI-Index

CMSI	CMSI-status
C_1	FALSE
C_2	FALSE
C_3	TRUE
:	:
C_i	TRUE
:	:
C_k	FALSE
:	:
C_b	TRUE

HSS Database

IMSI	C	CNEW	K
$IMSI_1$	C_1	$CNEW_1$	K_1
$IMSI_2$	C_2	$CNEW_2$	K_2
:	:	:	:
$IMSI_i$	C_i	$CNEW_i$	K_i
:	:	:	:

Fig.1. The HSS's database and the CMSI-Index.

A pool of base b $=2^{34}$ unique CMSI entries called CMSI-Index is stored in the HSS (Fig.1). Each CMSI entry in the CMSI-index has a value called CMSI-status against it. A CMSI that is already allocated to some UE will have FALSE in its CMSI-status indicating that this CMSI is not available for use. A CMSI-status having TRUE against a particular CMSI in the CMSI-index indicates that the CMSI is not used by any UE and is available for use. Finally, an operator specific function ENC is used in HSS to encrypt CNEW and the sequence number SQNHE using the key SQN to produce the encrypted challenge RAND.

B. The UE

The USIM of the UE is also extended to store a unique CMSI value that the UE shall transmit when it is requested to present its IMSI. Before the first connection a unique CMSI value called CMSIFIRST is embedded into the USIM by the service provider. The CMSIFIRST value is also stored in the HSS database in CNEW against the USIM's IMSI and the status of the CMSIFIRST entry in the CMSI-Index is set to FALSE. The CMSIFIRST is used only once during the initial run of the enhanced AKA protocol. Finally, an operator specific function DEC is used at the UE to decrypt a challenge RAND and extract the CNEW and the SQNHE using the random key SQN included in AUTN.

C. The Enhanced AKA Protocol

Whenever a UE wishes to connect to the network, it transmits to the MME an attach request that includes its CMSI (in the initial connection the attach request shall include the CMSIFIRST). Upon receiving the request, the MME transmits an authentication data request along with the received CMSI to the HSS. A new CMSI is generated at the HSS and supplied to the MME, which in turn forwards the CMSI to the UE. The enhanced AKA protocol is illustrated in Fig.2 and discussed in details below.

C.1. The HSS

On receiving the request message, the HSS generates authentication vector AV as follows:

1. Verify that the arriving CMSI is currently in use by some UE. If CMSI is not-in-use, the request is rejected.
2. Update CMSI and other UE's related information at the HSS if CNEW was transmitted by the UE.

2.1. Select a fresh CMSI(CFRESH) from CMSI-Index

update CFRESH ← CMSI-Index

2.2. Free up the old CMSI C

update CMSI-Index ← C

2.3. Update the C and CNEW stored against the UE's IMSI at the HSS database.

update C ← CNEW
update CNEW ← CFRESH

2.4. Update the sequence number SQNHE.

update SQNHE ← SQNHE + 1

3. Generate a new random key SQN

$$SQN = \{0, 1\}^{48}$$

4. Compute challenge RAND by encrypting CNEW and SQNHE using ENC with SQN as the input key

RAND = ENC (SQN, (m = (CNEW, SQNHE)))

5. Compute MAC over SQN, AMF, and RAND MAC = f1(K, (SQN, AMF, RAND)
6. Compute the remaining authentication parameters: XRES, CK, IK, AK, and AUTN.
7. Transmit the authentication vector AV to MME that shall forward AUTN and RAND to the UE.

The changes made in the standard AKA procedure for the generation of authentication vector can be summarized as follows:

- Step 1 (new step) for checking the validity of the arriving CMSI.
- Step 2 (new step) for updating CMSI and other UE's related information at the HSS if necessary.
- Step 3 (modified step) SQN is randomly generated rather than being computed as in the standard procedure.
- Step 4 (modified step) RAND is computed rather than being randomly generated as in the standard procedure.
- Other steps remain unchanged.

Before to decide whether to accept a CMSI based authentication request or not, the HSS must first verify that an arriving CMSI is valid and currently in use by some UE. This is done by locating the incoming CMSI in the HSS's database (step 1). If no match is found, the request is rejected. If a match is found, the HSS locates the corresponding UE's IMSI and the secret key *K*. In step 2, the HSS verifies that the incoming CMSI is the

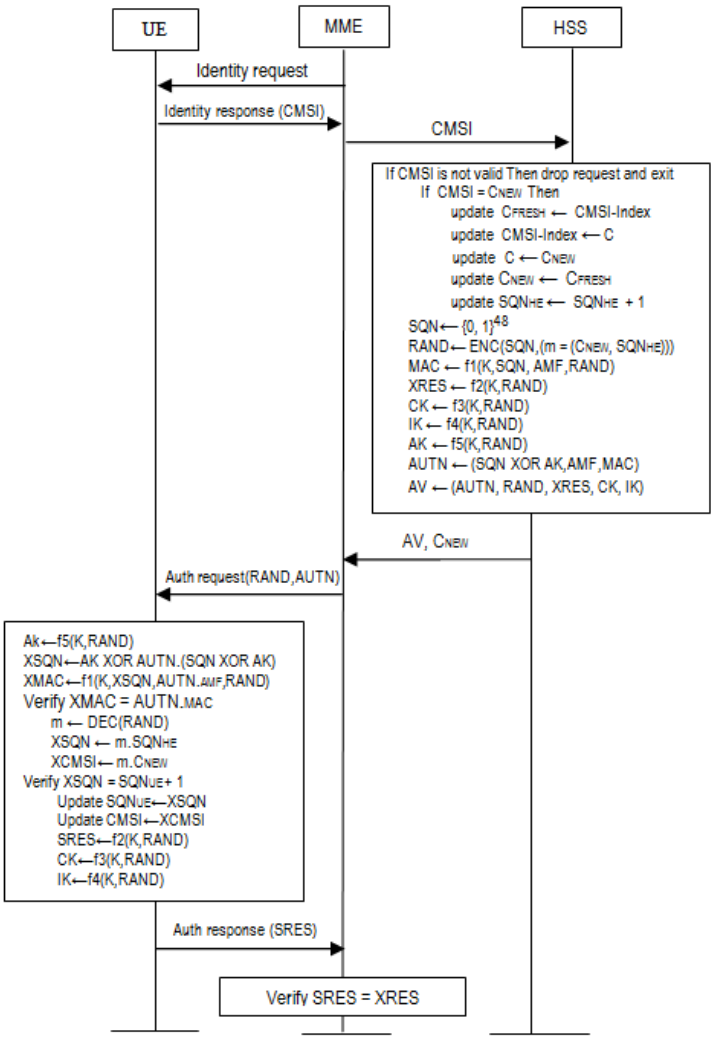

Fig.2. The Enhanced AKA Protocol.

latest CMSI transmitted to the UE. It checks CMSI=CNEW. If they match, the HSS allocates a fresh not-in-use CMSI called CFRESH to the concerned UE and updates information relating to the UE at HSS like the CMSIs, and the sequence number SQNHE (sub-steps 2.1 through 2.4).

Handling of sequence number at the HSS, to some extent, is different from the standard AKA. The HSS updates SQNHE only after it confirms that the past authentication event was successful after receiving CNEW from the UE. The benefits of this scheme are: (1) the synchronization between the HSS and the USIM is always maintained. At any point of time, after a successful authentication event, SQNHE and SQNUE each store the same value. Losing AVs in transit will not affect the synchronization since the values of SQNHE and SQNUE remain synchronized, (2) the HSS cannot be forced out-of-sync by an attacker who sends multiple authentication requests with CMSI that was previously used by some UE. Similarly, the UE cannot be forced out-of-sync due to a hacker who resends intercepted AUTN and RAND to the UE, (3) it is easy for the UE to detect a replay attack mounted by a hacker, who resends AUTN and RAND to the UE, and (4) authentication failure due to so-called synchronization failure will not occur. For successful functioning of this scheme, the MME (as Ref. [1] recommends) should fetch only one AV from the HSS at a time.

A random key SQN is generated in step 3 using an operator-specific function. However, the operator can use the function f0 in standard AKA procedure to generate random SQN with the desirable length. The HSS shall use the SQN as the input key of encryption function *ENC* for protecting the confidentiality of challenge RAND. Besides being a random number, SQN is also protected by a safe key AK. Hackers are unable to obtain SQN.

Similarly, the computation of RAND (step 4) was changed from generating a random number in the standard procedure. RAND (128 bits) is produced from encrypting CNEW and SQNHE using the random key SQN. By utilizing RAND to provide the UE with CMSI and sequence number, no change is required to the system messages.

C.2. The UE

Upon receipt of RAND and AUTN the UE verifies the AUTN token and computes the authentication response message as follows:

1. Computes the anonymity key AK using f5
2. Extracts the random key XSQN
3. Then the UE computes XMAC over XSQN, RAND and AMF and compares XMAC with the MAC included in the AUTN.
4. If MAC verification passes, the UE decrypts RAND using function *DEC* with the input key XSQN to retrieve m.CNEW and m.SQNHE,
5. The UE verifies the received sequence number SQNHE.

Verify if m.SQNHE = SQNUE + 1

6. If the SQNHE is within the correct range, AUTN is verified successfully, the UE updates the sequence number SQNUE, and computes the response RES, the ciphering key CK and the integrity key IK.
7. Updates its CMSI to m.CNEW.
8. The response RES is transmitted back to the MME and the two keys are stored in the phone. The MME compares the response of the UE (RES) with the expected response XRES that was included in AV received from HSS. If they are equal, the authentication is successful, otherwise the MME sends an authentication reject message to the UE.

The steps 4 and 7 are introduced by the proposed protocol; the other steps remain the same. Step 4 is for decrypting RAND using function *DEC* in order to retrieve the new CMSI (m.CNEW) and the sequence number (m.SQNHE) while step 7 is for updating CMSI to the newly received CMSI (m.CNEW).

D. Formal Verification

To prove that the enhanced AKA protocol presented above preserves the user identity privacy and provides user anonymity and unlinkability of user, we run the ProVerif tool [14] on the enhanced AKA protocol. The enhanced AKA protocol is proved to hold the anonymity and unlinkability properties and the ProVerif code is given in the Appendix. The underlying idea behind the proof is that an attacker (outside observer) sees no difference in the output of two executions of protocol that they differ only in user identities. Details of how anonymity and unlinkability are defined using the ProVerif can be found in Ref. [9].

V. ANALYSIS

In this section, we present the key features and the security analysis of the proposed solution, compare our solution against the standard AKA protocol as well as against related work.

A. The Key Features

1) ***Minimal Computation Overhead***: This solution places the majority of computation overhead on the HSS while a minimal computation is placed on the UE. Since the computation power of the HSS is unlimited, we claim that the overhead is negligible. We also claim that the computation overhead at the UE is negligible.

2) ***Minimal system Impact:*** The solution does not require a changes in the messages and the messaging system, which makes it transparent to the intermediary networks.

3) ***Compatible with LTE architecture:*** The solution can fit easily in the current architecture as it imposes minimal modifications on the network parties.

B. Security Analysis

In this section, the security of the proposed solution is analyzed and its resistance against several attacks is studied.

1) ***Identity Privacy Protection:*** The solution solves the exposure problem of the IMSI through the use of ever changing temporary identifiers CMSIs. The IMSI is always kept secret between the home network (HE) and the UE and no entity else in the network knows about it. During the USIM's lifetime, the IMSI never leaves the HSS database and the USIM and is never used in any networking operation.
2) ***Replay Attack:*** The solution protects user against replay attack. Assume that an attacker has intercepted an authentication vector AV (AUTN and RAND) destined to a particular user during a successful run of enhanced AKA protocol. If the attacker attempts to resend the AV to the user then the user will easily detect such attack by comparing the sequence number stored at the UE with the sequence number included in the RAND token. In such case, the received sequence number would be less than the sequence number stored at the UE.
3) ***Identity Guessing:*** The number of total possible CMSIs $T = 2^{34}$ and the probability that an attacker guesses a true CMSI $P = 1/T$. As T is a large number, it is obvious that the probability of correctly guessing a particular user's CMSI is negligible.
4) ***User Anonymity:*** The user identity is an important aspect of user privacy. The proposed solution provides a high level assurance of preserving user identity. It is obvious that an attacker cannot know IMSI of a particular UE since it is accessible only to the HSS and the UE (USIM) and no other party in the network knows about it; the IMSI never used or transmitted. In regard to the CMSI, no way for an attacker to know the CMSI assigned to a particular UE because the CMSI is encrypted by the HSS before sending it to the UE. The CMSI of a UE is kept hidden from the attacker until it is used by the UE for the identification purpose. It is worthy to be mentioned that the attacker cannot get an advantage of knowing a particular CMSI.

The strategy adopted by the proposed solution with respect to CMSI selection and use gives a user a privilege to preserve the user anonymity and prevents attackers from breaching the user anonymity. Because CMSI is utilized only once by the UE. Once the UE is successfully identified by the network, it is assigned a new CMSI which is different from the current CMSI that was last used. The new CMSI being assigned to the UE is random and is unrelated to the most recently CMSI used by the UE. From the point of view of an attacker, CMSIs assigned to a particular UE look as random bit streams that cannot be linked to a certain UE. As a result, the attacker cannot identify the target subscriber and the highest level of identity anonymity is provided.

5) ***User Untraceability:*** Traceability refers to the possibility of identifying past of identity requests and responses of the same subscriber. The solution eliminates user traceability and protects user against tracking attack through introducing of pseudonyms that replaces the user permanent identifier (IMSI). The solution assigns the user a different CMSI every time the user attaches to the network. CMSIs assigned to a particular user cannot be distinguished nor linked to each other by an outside observer, and thus untraceability of the user is maintained.
6) ***Unlinkability:*** the solution provides unlinkability of LTE network subscribers. As the CMSI identifier is utilized only once by the UE, this makes it difficult for an observer to identify the identity requests and responses destined the same user. From the observer's view point the CMSIs exchanged in the network are random and unrelated. Consequently, the observer cannot identify the past identity requests and responses of the same user, and the unlinkability of the user is provided.
7) ***DoS Attack:*** The attacker cannot mount DoS "Denial of service" attack against the network by sending multiple attach requests repeating the use of legitimate CMSI. As a user consumes a particular CMSI only once before it is replaced by the HSS, the network does not expect to receive multiple attach requests parameterized with the same CMSI. The received attach requests with the same CMSI are discarded by the network.

In addition, there is no way for an attacker to force the HSS to be out-of-sync as the HSS increments the sequence number only once it receives a new CMSI (CNEW), which was confidentially transmitted to the UE and still not utilized by the UE and not known by the attacker as well. Similarly, the UE cannot be forced out-of-sync due to a hacker who resends intercepted AUTN and RAND to the UE.

C. Our Solution Vs. Standard AKA Protocol

Compared with existing standard AKA protocol of LTE, the enhanced AKA protocol is proved to provide an additional security performance as follows:

1) Via replacing the permanent identity IMSI with pseudonyms, the proposed protocol can effectively avoid the problems that may result as a consequence of the exposure of the IMSI;
2) The protocol has adopted an independent sequence number (SQN) management mechanism that is proved to maintain the synchronization between the network parties and eliminate the probability that any network party (the HSS and the UE) get out-of-sync. In fact, the proposed mechanism of sequence number management prevents network errors due to synchronization failure. As the sequence numbers

maintained by both parties participating in AKA procedure are incremented only during a successful run of AKA, both sequence numbers remain synchronized all the times.

3) Changing the nature and functionality of existing SQN parameter adds security to the current AKA protocol. Generating SQN randomly and utilizing SQN as a key that is used only once for protecting the new pseudonym CMSI that will be transmitted to the user, gives more protection to CMSI.
4) The protocol prevents the possibility of launching DoS "Denial of service" attack against the network. As a user consumes a particular CMSI only once before it is replaced by the HSS, the network does not expect to receive multiple attach requests parameterized with the same CMSI. As a result, the received attach requests with the same CMSI are discarded by the network.

D. Our Solution Vs. Related Work

In the following we compare the presented solution with very closely related research efforts. The comparison begins with approaches of the first category i.e., approaches for enhancing privacy that utilize public key cryptography. This category includes works of [7, 8, 9, 11]. The main characteristic of the works of this class is that they utilize public key cryptography to encrypt the IMSI.

The main difference between above works and our solution in regarding to protecting user identity privacy is a philosophical one. The approaches require that the UE to perform encryption operation before sending IMSI and that the HSS decrypts the encrypted IMSI before processing the request. The anonymity of the UE would be preserved during the attach procedure. Although public key cryptography offers anonymity of the user, the high computation complexity and communication overhead as a result of transmitting encrypted IMSIs from one MME to another in handover and authentication procedures are still considerable. Our philosophy is that the design of the approach must consider the computational power of the UE must keep the processing efforts bearable at the UE.

Authors in [2] address problem of enhancing user identity privacy in LTE. The proposed scheme is based on what is called a Dynamic Mobile Subscriber Identity (DMSI), which is composed of *MCC||MNC||RIC||ERIC*. The *RIC* and *ERIC* are a random identity confidentiality number and the encrypted RIC. Ostensibly, the *RIC* and *ERIC* both need to be 128 bit wide. The *RIC* (and *ERIC*) are dynamically assigned and the association with IMSI is unknown to external parties. Our solution is very close to theirs, however, their approach is relatively complex, with several new cryptographic functions. The management of DMSIs would need additional processing effort and memory cost. In addition, the amount of computational effort for mobile devices is high since DMSI is changing with every authentication procedure.

In [3] the authors suggest concealing the real IMSI within a random bit stream of certain size where only the subscriber and HSS could extract the respective IMSI. The main differences are: (1) their solution needs a considerable modification on the architecture while our solution needs minimal modifications and can fit easily within the architecture; (2) unlike their solution, in our solution the IMSI is never transmitted, and thus its confidentiality is guaranteed.

Our presented solution is similar to the solution presented in [13] in that the IMSI is replaced with a changing pseudonym that only the HSS can link to the UE's identity. The hiding of the IMSI is done without changing any of the system messages. During authentication, the HSS supplies the UE with a random new Pseudo Mobile Subscriber Identifier. Both solutions propose to use the random challenge (RAND) to provide the UE with the Pseudonymous.

Despite the above similarities, there are some differences.

(a) Their solution requires that HSS server and each UE must be extended to store three additional values: the new shared secret key κ and the two PMSI values p and p′. Here p is used to store the PMSI value the UE is currently using and p′ stores the new PMSI value that the HSS server designates as the successor PMSI for that UE. Our solution requires the HSS to store only two additional values for each UE: C and CNEW and the UE to store only one additional value: CMSI. As the solution in [13] requires storing more additional values at the HSS and the UE, it is obvious that the memory cost in [13] is higher.

(b) The solution in [13] requires that HSS and each UE must have an additional shared secret key κ to encrypt PMSI. The compromising of the key κ will lead to compromising the entire method. Instead, our solution utilizes the SQN token as a key to encrypt and hide the CMSI to be transmitted to the UE. A new SQN key is randomly generated each time the enhanced AKA is run. Using SQN as a key has an advantages: (1) no additional key is required to conceal the key; (2) as SQN is changing key, the threats of breaching the privacy by knowing SQN value is minimized.

(c) Another difference with [13] lies in the sequence number management. Their approach always increases the sequence number at HSS, which would enable an adversary to mount denial of service attack DoS against the HSS. In fact, by repeating sending PMSI included in multiple fake attach requests to the HSS, an adversary can enforce the server HSS to be out-of-sync. Conversely, our approach eliminates this possibility as it increases the sequence number at both HSS and UE only in the event of successful run of enhanced AKA protocol.

VI. CONCLUSION

This paper presents a convenient solution to the

problem of protecting user identity privacy of the users in LTE network. The identity privacy is maintained through a secure identification scheme that allows a user to be uniquely identified by the home network (HE) while the user remains anonymous within the network, and thus prevents adversaries from being able to identify a user. The solution derives its advantages from the fact that it is compatible with current standards of LTE cellular technology and easily fits within the current architecture. The presented solution preserves the user identity privacy in LTE with minimal modifications at both the network and the UE and low computation overhead on the part of the network and negligible computation overhead on the part of the UE.

Besides user identity privacy issues, the privacy issues related to paging procedure and location tracking represent serious threats to user privacy. They must not be overlooked. Our future work is to solve paging and location tracking issues and investigate how the solutions can be integrated.

APPENDIX A FORMAL VERIFICATION OF ENHANCED AKA PROTOCOL

The main result of this appendix is that the presented enhanced AKA protocol indeed enforces secure identification and authentication and preserves user identity privacy (i.e., user anonymity). The underlying idea behind the proof is that an attacker (outside observer) sees no difference in the output of two executions of protocol that differ only in user identities. The proof proceeds by using observational equivalence.

Enhanced AKA Protocol:

```
type key.
fun enc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key; dec(enc(m, k), k) = m.
fun tc(bitstring): key [data, typeConverter ].
fun select(bitstring, bitstring, bitstring, bitstring, bitstring):
bitstring
   reduc forall a: bitstring, b: bitstring, c: bitstring, d:
bitstring; select(a, a, b, c, d) = b
   otherwise forall a: bitstring, b: bitstring, c: bitstring, d:
bitstring; select(a, b, c, a, d) = d.
fun xor_enc(bitstring, key): bitstring.
reduc forall m: bitstring, k: key; xor_dec(xor_enc(m, k), k)
= m.
fun getKey(bitstring): key [private].
fun getSQN(bitstring): bitstring [private].
fun next(bitstring): bitstring.
fun f1(key, bitstring, bitstring, bitstring): bitstring.
fun f2(key, bitstring): bitstring.
fun f3(key, bitstring): key.
fun f4(key, bitstring): key.
fun f5(key, bitstring): key.

free s2s: channel.
free s2h: channel [private].

const ID_REQUEST: bitstring.
const ID_RESPONSE: bitstring.
const AUTH_REQUEST: bitstring.
const AUTH_RESPONSE: bitstring.

free amf: bitstring [private].
free A: bitstring.
free B: bitstring.

let MME =
out(s2s, ID_REQUEST);
in(s2s, (=ID_RESPONSE, cmsi: bitstring));
out(s2h, cmsi);
in(s2h, (rand: bitstring, (sqn_ak: bitstring, amf: bitstring,
mac: bitstring), xres: bitstring, ck: key, ik: key));
let autn = (sqn_ak, amf, mac) in
  out(s2s, (AUTH_REQUEST, rand, autn));
  in(s2s, (=AUTH_RESPONSE, =xres)).
let HSS
  (id1: bitstring, cmsi1_old: bitstring, cmsi1_new:
   bitstring, sqn1: bitstring, id2: bitstring, cmsi2_old:
   bitstring, cmsi2_new: bitstring, sqn2: bitstring) =
      new ksqn:bitstring;
      in(s2h, cmsi_in: bitstring);
      let cmsi_new = select(cmsi_in, cmsi1_old,
          cmsi1_new, cmsi2_old, cmsi2_new) in
      let k = getKey(select(cmsi_in, cmsi1_old, id1,
              cmsi2_old, id2)) in
      let sqn = select(cmsi_in, cmsi1_old, sqn1,
              cmsi2_old, sqn2) in
      let rand = enc((cmsi_new, sqn), tc(ksqn)) in
      let mac = f1(k, sqn, AMF, rand) in
      let xres = f2(k, rand) in
      let ck = f3(k, rand) in
      let ik = f4(k, rand) in
      let ak = f5(k, rand) in
      let autn = (xor_enc(ksqn, ak), AMF, mac) in
      out(s2h, (rand, autn, xres, ck, ik)).
let UE(id: bitstring, cmsi: bitstring, sqn: bitstring) =
      let k = getKey(id) in
      in(s2s, =ID_REQUEST);
      out(s2s, (ID_RESPONSE, cmsi));
      in(s2s, (=AUTH_REQUEST, rand: bitstring,
        (sqn_ak: bitstring, amf: bitstring, mac: bitstring)));
      let ak = f5(k, rand) in
      let ksqn = xor_dec(sqn_ak, ak) in
      let (cmsi_new: bitstring, xsqn:bitstring) = dec(rand,
          tc(ksqn)) in
      if xsqn = sqn && mac = f1(k, xsqn, amf, rand) then
          let sres = f2(k, rand) in
          let ck = f3(k, rand) in
          let ik = f4(k, rand) in
          out(s2s, (AUTH_RESPONSE, sres)).
process
      new cmsi1a: bitstring; new cmsi2a: bitstring;
       new cmsi3a: bitstring;
      new cmsi1b: bitstring; new cmsi2b: bitstring;
       new cmsi3b: bitstring;
      new sqn1a: bitstring; new sqn1b: bitstring;
      (
      !UE(A, cmsi1a, sqn1a) | !UE(B, cmsi1b, sqn1b) |
```

```
!UE(choice[A, B], choice[cmsi2a, cmsi2b],
      choice[next(sqn1a), next(sqn1b)]) |
!MME |
!HSS(A, cmsi1a, cmsi2a, sqn1a, B, cmsi1b,
        cmsi2b, sqn1b) |
!HSS(A, cmsi2a, cmsi3a, next(sqn1a), B, cmsi2b,
        cmsi3b, next(sqn1b))
```

REFERENCES

[1] 3GPP, 3GPP System Architecture Evolution (SAE); Security architecture. 3GPP, TS 33.401, 2013.

[2] Choudhury H., Roychoudhury B. and Saikia D. K., Enhancing user identity privacy in LTE. In IEEE 11th International Conference on Security and Privacy in Computing and Communications (TrustCom), 2012. p. 949–957.

[3] Hamidreza Ghafghazi, Amr El-Mougy, Hussein T. Mouftah, Enhancing the Privacy of LTE-based Public Safety Networks. In 13th Annual IEEE Workshop on Wireless Local Networks, Edmonton, Canada 2014.

[4] Bikos A. and Sklavos N., LTE/SAE security issues on 4g wireless networks. IEEE Security and Privacy, 2013. 11(2):p. 55–62.

[5] Seddigh N., Nandy B., Makkar R. and J. F. Beaumont H. F., Security advances and challenges in 4g wireless networks. In Eighth Annual International Conference on Privacy Security and Trust (PST), 2010. p. 62-71.

[6] Bilogrevic I., Jadliwala M. and Hubaux J. P., Security and privacy in next generation mobile networks: LTE and femtocells. In 2nd International Femtocell Workshop, Luton, UK. Citeseer, 2010.

[7] Bou A. J., Chaouchi H. and Aoude M., Ensured Confidentiality Authentication and Key Agreement Protocol for EPS. In 3rd Symposium on Broadband Networks and Fast Internet, 28-29 May 2012.

[8] Xiehua, Li, and Wang Yongjun, Security Enhanced Authentication and Key Agreement Protocol for LTE/SAE Network. In 7th International Conference on Wireless Communications, Networking and Mobile Computing, IEEE, 2011.

[9] Arapinis M., et al., New privacy issues in mobile telephony: fix and verification. In ACM Conference on Computer and Communications Security, 2012.p. 205–216.

[10] Muxing Z., Yuguang F., Security Analysis and Enhancements of 3GPP Authentication and Key Agreement Protocol," IEEE Trans, vol. 4, 2005.p. 734-742.

[11] Køien G. M., Mutual entity authentication for LTE. In 7th International Wireless Communications and Mobile Computing Conference, IEEE, 2011.

[12] Køien G. M., Privacy enhanced mutual authentication in LTE. In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), IEEE, 2013. p 614–621.

[13] Fabian van den Broek, Roel Verdult and Joeri de Ruiter, Defeating IMSI Catchers. In CCS '15 Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM New York, NY, USA 2015.

[14] B. Blanchet. Proverif: Cryptographic protocol verifier in the formal model. http://www.proverif.ens.fr/.

[15] Kadhim Shubber for Wired magazine. Tracking devices hidden in London's recycling bins are stalking your smartphone. http://www.wired.co.uk/news/ archive/2013-08/09/recycling-bins-are-watching-you. Last accessed May 2015.

[16] Myrto Arapinis, Loretta Ilaria Mancini, Eike Ritter, and Mark Ryan. Privacy through pseudonymity in mobile telephony systems. In NDSS, 2014.

[17] Siraj Datoo for The Guardian. How tracking customers in-store will soon be the norm. http://gu. com/p/3ym4v/sbl. Last accessed May 2015.

[18] Balasaheb N. Jagdale,Nileema S. Gawande,"Hybrid Model for Location Privacy in Wireless Ad-Hoc Networks", IJCNIS, vol.5, no.1, pp.14-23,2013.DOI: 10.5815/ijcnis.2013.01.02

[19] Stuart Owen Goldman, Richard E Krock, Karl F Rauscher, and James Philip Runyon. Mobile forced premature detonation of improvised explosive devices via wireless phone signaling. US Patent 7552670, June 30 2009.

[20] Michael B¨ock. Simulation chamber and method for setting off explosive charges contained in freight in a controlled manner. US Patent 14345697, September 19 2012.

Authors' Profiles

Abdulrahman A. Muthana received his B.Sc in Computer Science from Mosul University, Iraq, M.Sc in Computer Applications from Bangalore University, India and PhD in Information Security from University Putra Malaysia in. His research areas include information security, smartphone security, network security, software security. He is now an Assistant Professor in Faculty of Computer Science and Information Systems, Thamar University, Yemen.

Mamoon M. Saeed received his Bachelor degree in Electrical and Telecommunication Engineering from Sana'a University, Yemen. Recently, he is a master student at department of Computer Networks and Information Technology in Yemen Academy for Graduate Studies. His research areas include information security, Telecommunication security, and network security.

Manuscript received January 16, 2016; revised June 11, 2016; accepted July 21, 2016.

How to cite this paper: Abdulrahman A. Muthana, Mamoon M. Saeed,"Analysis of User Identity Privacy in LTE and Proposed Solution", International Journal of Computer Network and Information Security(IJCNIS), Vol.9, No.1, pp.54-63, 2017.DOI: 10.5815/ijcnis.2017.01.07

